

## Das Siemens-PEARL-Compilersystem

Von Dipl.-Math. H. Schoknecht, Karlsruhe

### 1. Implementierter Sprachumfang

Der implementierte Sprachumfang umfaßt Basis PEARL mit folgenden zusätzlichen Erweiterungen, die alle Sprachelemente aus Full PEARL sind.

- Initialisierung von Feldern und Strukturen
- Mehr als drei Dimensionen für Felder
- Untere Dimensionsgrenze von Feldern beliebig angebar (auch negativ)
- Zuweisung kompletter Felder und Strukturen
- Mehrfachzuweisung ist erlaubt
- Operatoren LWB, UPB auch monadisch
- Bit- und Charakter-Gruppenselektion (auch auf linker Seite einer Zuweisung)
- Bei Taskoperationen ist Liste von Einplanungen erlaubt
- ACTIVATE mit Prioritätsangabe
- Modulo-Funktion (REM) ist implementiert
- Bei Stringzuweisung wird auch abgeschnitten (mit Ausgabe einer Warnung)
- TOFICED für CHAR (2)
- TOCHAR für FIXED (7)
- B2-Format ist erlaubt
- Für folgende Kurzformen sind auch entsprechende Langformen zulässig

CHAR	CHARACTER
DCL	DECLARE
DUR	DURATION
IDENT	IDENTICAL
INIT	INITIAL
IRPT	INTERRUPT
PRI0	PRIORITY
PROC	PROCEDURE
SPC	SPECIFY

### 2. Kurzdarstellung der verwendeten

#### Compiler-Technologie

#### 2.1 Einbettung in die System-Software

Im Hinblick auf den späteren Einsatz des PEARL-Compilers als Produkt wurde von Anfang an Wert darauf gelegt, ihn voll in das Spektrum der Soft-

ware-Erstellungsmittel der Rechnerfamilie 300/16 Bit zu integrieren (/1/, /2/, /3/). Dies führte insbesondere zu folgenden Zielvorstellungen:

- Programmierung des PEARL-Compilers in den verfügbaren und den Wartungsdienststellen vertrauten Programmiersprachen Assemblersprache ASS300 und Syntaxanalyseprache MECO 300.
- Erzeugung von Grundsprache GS300, die für die Weiterverarbeitung durch Binder und Lader geeignete Darstellung der Maschinensprache der Rechnerfamilie 300/16 Bit.
- Übersetzung von Moduln aus Quell- in Grundsprachebibliotheken unter Einhaltung der Modul- und Bibliotheksstruktur, wie sie von den vorhandenen Dienst- und Hilfsprogrammen der Rechnerfamilie 300/16 Bit verlangt wird.
- Möglichkeit der Kopplung der vom PEARL-Compiler erzeugten mit anderen GS300-Moduln.
- Ablauffähigkeit des Compilers als Hintergrundprogramm in einem beschränkten Laufbereich von 17 KW (Wortlänge 16 Bit).
- Ablauffähigkeit sowohl des Compilers als auch der von ihm übersetzten Programme unter dem Standard-Organisationsprogramm ORG300.

#### 2.2 Förderung der Akzeptanz bei Anwendern

Nicht weniger wichtige Kriterien für den Compiler-Entwurf ergaben sich aber auch aus der Notwendigkeit, eine positive Einstellung der Anwender zum neuen Erstellungshilfsmittel PEARL nicht von vornherein durch ausschließlich compilerbedingte Unzulänglichkeiten zu verhindern. Dies führte zu folgenden Forderungen:

- Hohe Kompiliertgeschwindigkeit trotz der Laufbereichsbeschränkung.
- Ausführlichkeit und Genauigkeit der Compilermeldungen im Hinblick auf Art und Lokalisierung eines entdeckten Fehlers.
- Gute Effizienz des generierten Objektcodes durch volle Ausnutzung des Befehlsvorrats der Rechnerfamilie 300/16 Bit, aber ohne zusätzlichen Optimierungspaß.
- Integration leistungsfähiger Laufzeit-Testhilfen in das PEARL-Kompilersystem.
- Fehlererkennung möglichst zur Compilezeit (strenge Typprüfung).
- Gute Behandlung von inneren und äußeren Ereignissen (Alarme, Signals).

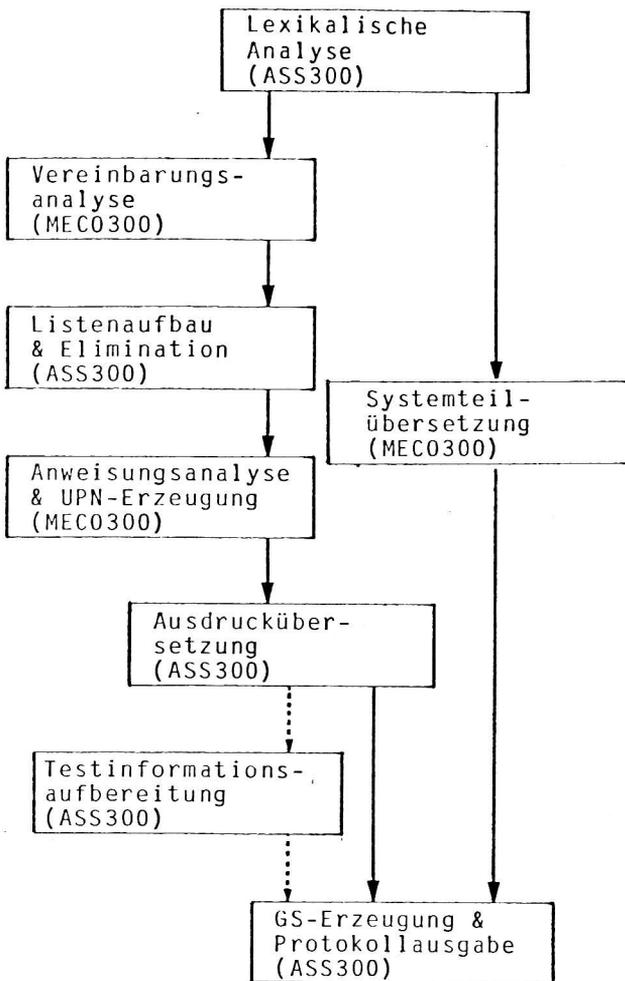


Bild 1: Pässe des PEARL-Compilers

### 2.3 Besonderheiten der Realisierung

Der Siemens PEARL-Compiler PC30 wurde als 8-Paß-Compiler realisiert (siehe Bild 1) und umfaßt in Kurzform folgende Besonderheiten (/4/, /5/):

- 17 KW Laufbereich.
- Lexikalische Analyse durch einen endlichen Automaten.
- Aufteilung Syntax- und Semantikanalyse auf vier Pässe: Bearbeitung von Systemteil, Deklarationen, Anweisungen und Bezeichnerlimitation. Der Bezug zwischen den Pässen wird durch eine gemeinsame Zwischensprache hergestellt, die wenig zusätzliche Tabellen benötigt. Syntaxanalyse erfolgt nach der Methode der syntaktischen Funktionen (Glennie'sche Syntaxmaschine). Vermeidung von Sackgassen durch zusätzliche bottom-up-Elemente.
- Codegenerierung erfolgt auf der Basis einer Stackmaschine, deren zwei oberste Elemente in zwei Registersätzen geführt werden. Grundsätzlich wird der rechte Operand vor dem linken ausgewertet.
- Der letzte Paß umfaßt die Grundsprache- und die Protokollausgabe. Bei möglichen Fehlern werden diese an Ort und Stelle mit umfassenden Texten beschrieben.

- Umsetzen der PEARL-Quellsprache direkt in Grundsprache (300-700 Quellsprachezeilen pro Min.)..

### 3. Existierende Komponenten des Siemens-PEARL-Kompiliersystems

#### 3.1 Struktur des Siemens-PEARL-Kompiliersystems

Die Struktur des Kompiliersystems ist vor allem auf die Forderung nach hoher Übersetzungsgeschwindigkeit unter gleichzeitiger Beschränkung des verfügbaren Laufbereichs ausgerichtet. Deshalb wurde der Compiler von vornherein als Mehrpaßcompiler konzipiert. Je nach dem, ob es sich um den System- oder den Problemtteil eines PEARL-Moduls handelt, wird die Quellsprache in 3 bzw. 6 Schritten in die Maschinensprache der Rechnerfamilie 300/16 Bit umgewandelt. Im Testmodus des Compilers wird bei der Übersetzung des Problemtteils zusätzlich ein siebenter Paß durchlaufen, der Testhilf Informationen aufbereitet. Der erste und der letzte Übersetzungsschritt stimmen jeweils in beiden Fällen überein, so daß der Compiler insgesamt 8 Pässe aufweist (siehe Bild 1, /6/, /7/).

#### 3.2 Einbettung in ein Kompiliersystem

Um PEARL-Programme erstellen und ablaufen lassen zu können, sind neben dem Compiler noch einige weitere Programme erforderlich (siehe Bild 2). Diese Programme sind jedoch nicht PEARL-spezifisch, sondern allgemein einsetzbare Dienstprogramme der Rechnerfamilie 300/16 Bit, nämlich

- der Texteditor (MEDIS) für das Schreiben und Korrigieren des PEARL-Quelltexts.
- Der Binder (BD30) für das Binden von PEARL-Moduln untereinander, mit Moduln des PEARL-Laufzeitsystemes und ggf. mit weiteren, nicht in PEARL geschriebenen Moduln, z.B. Assemblerprozeduren.
- der Lader eines (Standard-) Organisationsprogramms für das Laden der vom Compiler erzeugten GS-Moduln und das Herstellen von Adreßbeziehungen über Modulgrenzen hinweg.
- Der gesamte Erstellungsweg ist auch über Bedienmasken des Softwareerstellungssystems TESEUS möglich.

Für den Ablauf selbst wird neben einem leistungsfähigen Organisationsprogramm vor allem das PEARL-Laufzeitsystem benötigt. Es enthält, in Prozeduren eingebettet, alle die Befehlsfolgen und Daten, die vom Compiler wegen ihres Umfangs nicht unmittelbar in den generierten Code eingestreut, sondern in Form von Prozeduraufrufen angesprochen werden.

Die meisten Prozeduren des Laufzeitsystems sind ablauffinvariant und brauchen daher in einem Rechner nur einmal im Speicher zu stehen. Der leichteren Hantierbarkeit wegen wurden die meisten davon zu folgenden 8 Großmoduln zusammengefaßt:

- Grundbedarf, z.B. Routinen für Blockeintritt, Anmeldung einer Signalreaktion, Feldzugriff, usw.
- Mathematische Funktionen wie Sinus, Cosinus, usw.
- Kernroutinen für die Ein- und Ausgabe, insbesondere in binärer, unformatierter Form.

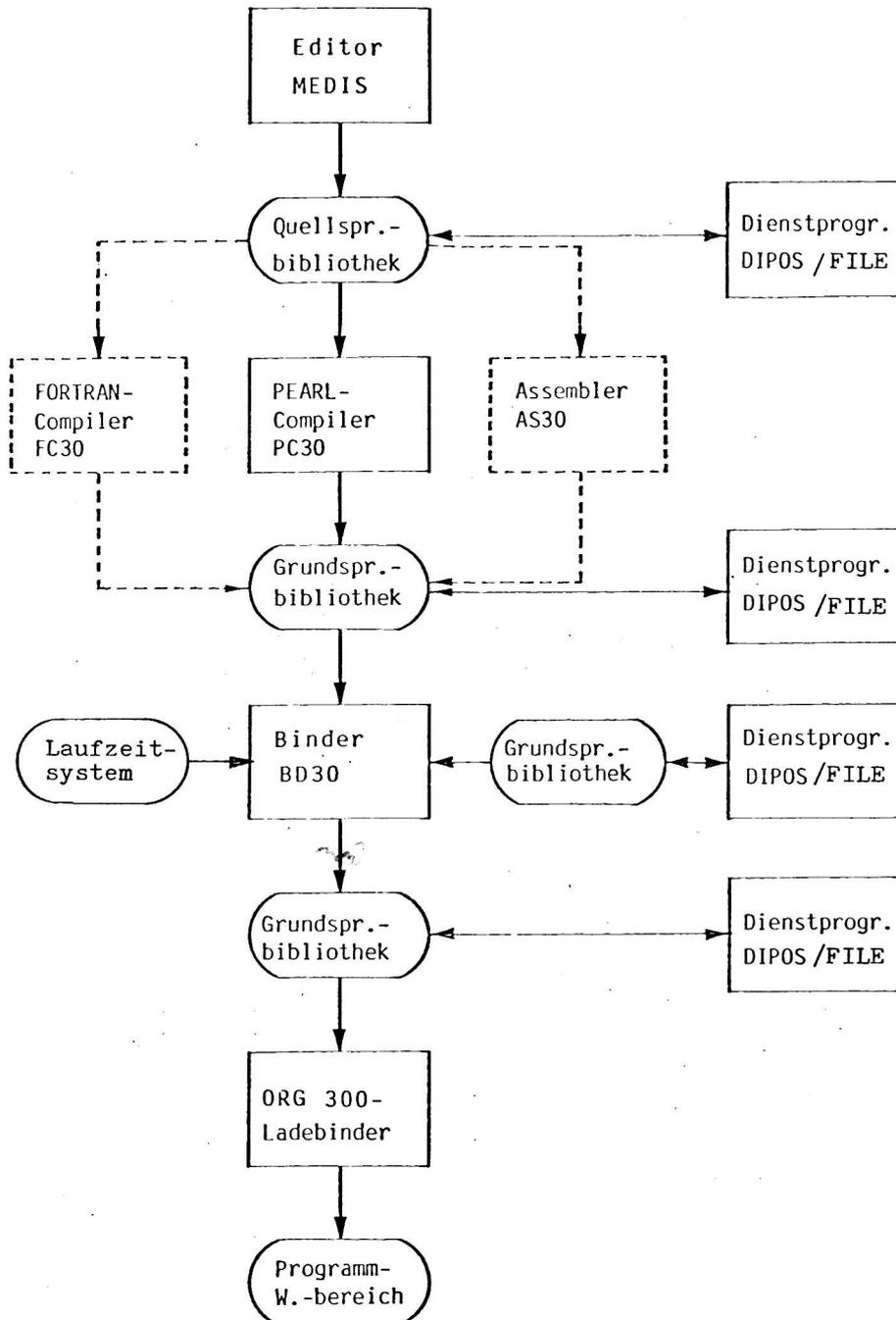


Bild 2: Erstellungsweg für PEARL-Programme

- Routinen zur Positionierung bei der Ein- und Ausgabe.
- Routinen zur Realisierung der GET-Anweisung.
- Routinen zur Realisierung der PUT-Anweisung.
- Routinen für die Ein- und Ausgabe in Dateien.

Diese Großmoduln können entweder zum PEARL-Programm hinzugebunden oder als Common Code geladen werden.

Neben den Großmoduln gibt es noch ca. 40 ablaufvariante kleine Treiberroutinen, die im allgemeinen nur zu den Tasks hinzugebunden werden, die sie benötigen; bei den wenigen nicht ablaufinvarianten

Laufzeitroutinen (z.B. der Taskanfangsroutine) ist dies Hinzubinden die einzige Möglichkeit, sie der aufrufenden Task verfügbar zu machen.

Für die gezielte Bearbeitung von Ereignissen stehen dem Anwender ca. 120 Standardsignale zur Verfügung, die in 4 Reaktions- und 8 verschiedene Fehlerklassen eingeteilt sind.

### 3.3 Testhilfen

Die vorhandenen leistungsfähigen Laufzeitesthilfen basieren auf den vom Testpaß aufbereiteten Informationen und umfassen folgende per Dialog ein- und ausschaltbare Funktionen:

- Protokollierung der Nummern durchlaufener Zeilen,
- Zeilenangabe für den Fehlerort bei evtl. auftretenden Laufzeitfehlern,
- unbedingtes Anhalten an wählbaren Zeilenanfängen,
- vom Wert einer Variablen abhängiges Anhalten an wählbaren Zeilenanfängen,
- Überprüfung und Änderung der Werte von Variablen.

Die Fehlermeldungen des Compilers erscheinen unmittelbar am Fehlerort im Quellspracheprotokoll eingestreut.

#### 4. Übersetzungsrechner und Zielrechner

Zwischen Übersetzungs- und Zielrechner muß nicht unterschieden werden, da eine Crossübersetzung nicht notwendig ist.

Ein Übersetzungs- oder Zielrechner kann sein:

Siemens 330, R10, R20, R30, R40

##### ● Ausstattung für Übersetzung

- Bediengerät
- Drucker
- Laufbereich für Compiler 17 KW
- Arbeitsbereich auf Platte mindestens 195 KW

##### ● Ausstattung für Zielrechner

- Standard- und Prozeßperipherie abhängig vom Anwendungsfall (insbesondere Bediengerät, Drucker, Platte, Grafik-Sichtgerät, Papierdatenträger, Prozeßsignalformer).

#### 5. Lieferform, Schulungsmaterial, Benutzerhandbücher, Wartungsleistungen

##### ● Lieferform

Der PEARL-Compiler wird als segmentiertes Programm in ladbarer Form zusammen mit der Library auf Platte ausgeliefert.

##### ● Schulungsmaterial

Für die Sprache PEARL existiert im Hause ein vierzehn Tage dauernder Kurs.

##### ● Benutzerhandbücher

Ein Benutzerhandbuch sowie eine Kurzbeschreibung für den versierten Anwender ist vorhanden (Best. Nr. P71100-D3010-X-X-35). Diese umfassen neben der Sprachbeschreibung auch die Bedienungsanleitung für den Compiler mit einer ausführlichen Fehler- und Signalbeschreibung.

##### ● Wartungsleistungen

Der Kunde hat eine 12 Monate dauernde Gewährleistung für das Funktionieren des Kompiliersystems.

#### 6. Referenzen und Einsätze

Die erste Produktfreigabe des PEARL-Kompiliersystems war im Januar 1978 (/8/). Seither erfolgten drei weitere Produktfreigaben, die aufgrund unserer Einsatzerfahrungen notwendig wurden und die gezielte Optimierungen und Erweiterungen gemäß Anwenderwünschen enthielten (/9/, /10/).

Referenzen Stand 5/80  
(Der PC 30 ist seit 1/78 lieferbar).

<u>Industrie:</u>	OBAG:	Regensburg
	Bayer AG:	Krefeld
	MBB:	Ottobrunn
	SDR:	Stuttgart
	GEW:	Köln
	Berufsförderungswerk:	Heidelberg
	Battelle:	Frankfurt
	BWB:	Eckernförde
	Verbundbergwerke	
	Haus Aden:	Oberaden

Hochschulen: Stuttgart, Berlin, Karlsruhe, Göttingen, Darmstadt, Dortmund

intern: Einsatz an mehreren Stellen

#### 7. Literaturhinweise

- /1/ Rieder, P.: Effiziente PEARL-Implementierung für den PR 330, Informatik-Fachberichte, Band 7, S. 173-183, Springer-Verlag, Berlin/Heidelberg/New York 1977.
- /2/ Degelow, L., Gottwald, H.-J., Schoknecht, H.: Stand der PEARL-Entwicklung. Tagungsbericht der 8. Jahrestagung des Siemens Anwenderkreises, S. 73-89, Fachhochschule Dortmund 1977.
- /3/ Schoknecht, H.: PEARL 300: Kompiliersystem und Verwendung für Echtzeitaufgaben. Tagungsbericht der 9. Jahrestagung des Siemens Prozeßrechner Anwenderkreises, Bericht KfK 2642, S. 161-175, Ges. für Kernforschung mbH, Karlsruhe 1978.
- /4/ Gottwald, H.-J., Schoknecht, H.: PEARL, eine leistungsfähige Echtzeit-Programmiersprache. Regelungstechnik S. 23-27 (1978).
- /5/ Dorn, M., Wenzel, T.: Prozeßsprache PEARL 300 für die Siemens-Systeme 300. Siemens-Zeitschrift, Band 52, S. 23-27 (1978).
- /6/ Siemens Erlangen (1980), Übersetzungsprogramme: PEARL 300. Best.Nr.: E-36/2205.
- /7/ Siemens Karlsruhe (1980), PC 30/PEARL 30: Compiler für PEARL, Programmbeschreibung. Best.Nr.: P71100-D3010-X-X-35.
- /8/ Bamberger, K.-F.: Das PEARL-Kompiliersystem für die Siemens 300-16 Bit. PEARL-Rundschau, Band 1, Nr. 1 S. 49-64 (1980).
- /9/ Struhulla, D.: Erfahrungen mit BASIC-PEARL im Projekt Netzleitstelle Deggendorf. PEARL-Rundschau, Band 1, Nr.2, S. 13-22 (1980).
- /10/ Weber, H.: Einsatz von PEARL bei der Software-Entwicklung für eine Fla-Schießplatz-Automatisierung. PEARL-Rundschau, Band 1, Nr.4, S. 26-31 (1980).